

## **Modul: Programmierung II**

**Semester:** 2

**Code der Veranstaltung:** PRG 2

**ECTS Punkte:** 5

**Kontaktstunden:** 40

**Selbststudium:** 85

**Dauer des Moduls:** 1 Semester

**Art des Moduls:** Pflicht

**Häufigkeit des Angebots:** jedes Studienjahr

**Gewichtung der Note in der Gesamtnote:** 5/180

**Art und Umfang der Prüfungsleistung:** KRS 90: Die Prüfung besteht entweder aus einer Klausur oder einem Referat oder einer Studienarbeit; im Fall einer Klausur gibt die Zahl den Umfang der Klausur in Minuten.

**Modulverantwortliche/r:**

Prof. Dr. Rohde, Prof. Dr. Ruckmann

**Lehrsprache:** Deutsch

**Dozenten:** Prof. Dr. Rohde, Prof. Dr. Ruckmann, Prof. Dr. Baeumle-Courth, Prof. Dr. Seifert

### **Zugangsvoraussetzungen:**

Programmierung I

### **Verwendbarkeit des Moduls:**

- Alle Module der Informatik

### **Qualifikations- und Kompetenzziele des Moduls:**

Die Studierenden kennen die Grundprinzipien des objektorientierten Programmierens. Sie verstehen und beherrschen die grundlegenden Konstrukte objektorientierter Programmiersprachen und können Programme auch höherer Komplexität zur Lösung konkreter Aufgaben und Probleme verstehen und selbständig entwickeln. Sie sind in der Lage, grundsätzliche Konzepte zur Modellierung objektorientierter Software anzuwenden. Sie haben außerdem einen exemplarischen Einblick in die Besonderheiten einzelner objektorientierter Programmiersprachen und deren Konstrukte gewonnen. Schließlich kennen sie grundsätzliche Prinzipien des Clean Code und können diese anwenden.

### **Lehr- und Lernmethoden des Moduls:**

Unterschiedliche Lehr-/Lernumgebungen: Präsenzveranstaltungen, Eigenstudium; Wechselnde Lehr-/Lernmethoden: Individuelles und kooperatives Lernen, problemorientiertes und integratives Lernen anhand von Übungen.

Zur Vermittlung der Inhalte wird eine moderne und etablierte, objektorientierte und stark typisierte Programmiersprache eingesetzt, z.B. Java, C++ oder C#.

## **Inhalt des Moduls:**

1. Grundprinzipien des objektorientierten Programmierens
  - 1.1 Klassen und Objekte, Attribute und Methoden
  - 1.2 Kapselung und Information Hiding
  - 1.3 Klassenhierarchie und Vererbung, Mehrfachvererbung, Kovarianz und Kontravarianz
  - 1.4 Polymorphismus
  
2. Spezifische Konstrukte objektorientierter Programmiersprachen
  - 2.1 Namespaces
  - 2.2 Interfaces
  - 2.3 Bibliotheken und Frameworks
  - 2.4 Ausnahme- und Fehlerbehandlung
  - 2.5 Fortgeschrittene Datentypen (z.B. Container, Iteratoren)
  - 2.6 Fortgeschrittene Sprachkonzepte (z.B. Attribute, Templates/Generics, Lambda-Ausdrücke)
  
3. Modellierung objektorientierter Software und Prinzipien des Clean Code
  - 3.1 Namenskonventionen und Coding Guidelines
  - 3.2 Entwurfsprinzipien, Kopplung und Kohäsion, SOLID-Prinzip
  - 3.3 Technische Schulden und Continuous Inspection
  - 3.4 Modellierung mit UML (nur UML-Klassendiagramme und einfache Beispiele)
  - 3.5 Entwurfsmuster (nur Konzept und einfache Beispiele wie Singleton oder Strategy)

## **Hinweis**

Die Modellierung von Software mit UML und Entwurfsmuster sind ebenfalls Inhalte in den Modulen „Software Engineering & Continuous Integration“ (alle Wirtschaftsinformatiker) sowie im Modul „Software Modeling & Architecture“ (Spezialisierung Software Engineering) und werden dort weiter ausgeführt.

## **Grundlegende Literaturhinweise:**

BALZERT, H., 2011. *Lehrbuch der Softwaretechnik. Entwurf, Implementierung, Installation und Betrieb*. 3. Aufl. Heidelberg: Spektrum Akademischer Verlag.

Je nach eingesetzter Programmiersprache, z.B.:

ALBAHARI, J. und B. ALBAHARI, 2018. *C# 7.0. kurz & gut*. 5. Aufl. Heidelberg: O'Reilly. O'Reillys Taschenbibliothek.

LOUIS, D., 2018. *C++*. München: Carl Hanser Verlag.

THEIS, T., 2018. *Einstieg in C# mit Visual Studio 2017. Ideal für Programmierneinsteiger*. 5. Aufl. Bonn: Rheinwerk Verlag GmbH. Rheinwerk Computing.

## **Ergänzende Literaturempfehlungen:**

GAMMA, E., R. HELM, R. JOHNSON und J. VLISSIDES, 2015. *Design Patterns. Entwurfsmuster als Elemente wiederverwendbarer objektorientierter Software*. Frechen: mitp.

OESTEREICH, B. und A. SCHEITHAUER, 2013. *Analyse und Design mit der UML 2.5. Objektorientierte Softwareentwicklung*. München: Oldenbourg.